Is DevOps a good career?

Manjunath.R
#16/1, 8th Main Road, Shivanagar, Rajajinagar, Bangalore560010, Karnataka, India
*Corresponding Author Email: manjunath5496@gmail.com
*Website: http://www.myw3schools.com/

Abstract

DevOps (a set of software development practices that combines software development (Dev) and information technology operations (Ops) to shorten the systems development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives) is becoming the standard way of working for Enterprises. Among the few powerful trends we had experienced in the recent times, one is undoubtedly the adoption of DevOps practices – and adoption of DevOps within the organization is rising on a broader scale, and Enterprises are trending toward it. DevOps builds upon best practices to help drive enterprise performance in modernizing environments. It offers organizations a new way to move the business forward and turn technology into a strategic advantage. An increasing number of businesses recognize the power that DevOps can bring a natural extension for Agile and continuous delivery approaches.

Patrick Debois is best known as the founder of DevOpsDays and as a creator of the DevOps movement, which explains why some refer to him as the "**Godfather of DevOps**".

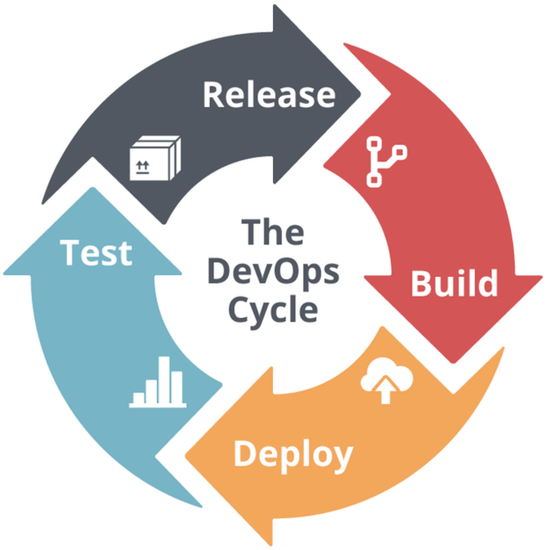
"At its essence, DevOps is a culture, a practice, a philosophy."

Introduction

DevOps expertise is in high demand. Job postings with "**DevOps**" in a title or keyword are sprouting up everywhere. DevOps is an **enterprise software development** phrase emerging from combination of IT teams, process and products to enable the continuous delivery of value to end users. It is a firm bond between **development** and **operations** that emphasizes a shift in mindset, better collaboration, and tighter integration and aims to create a culture and environment where building, testing, and releasing software can happen rapidly, often, and more reliably, so organizations can solve critical issues quickly, and better serve their customers and compete more effectively in the market.

What is DevOps?

"A software development method formed out of a fundamental need that stresses communication, collaboration and integration between software developers and **IT professionals**." DevOps could be explained simply as operations working together with engineers to get things done faster in an automated and repeatable way.



History of DevOps

At the 2008 Agile Toronto conference, **Andrew Shafer** and **Patrick Debois** introduced the term in their talk on "**Agile Infrastructure**". Since 2009, the DevOps term has been steadily promoted based on a simple philosophy — business works best when efforts being coordinated and collaborative — and brought into more mainstream usage through a series of "**DevOpsDays**", which started in Belgium and has now spread into Web-enabled sphere to resolve the conflict between the software developers and the operations teams when it comes to getting great work done quickly. In recent years, more tangential DevOps initiatives have also evolved, such as **OpsDev**, **WinOps**, and **BizDevOps** to encourage the communication between software developers and IT Operations to increase the speed at which applications being delivered.

Benefits of DevOps

The technical benefits include:

- Continuous software delivery
- Less complexity to manage
- Faster resolution of problems

The cultural benefits include:

- More productive teams
- Higher employee engagement
- Greater professional development opportunities

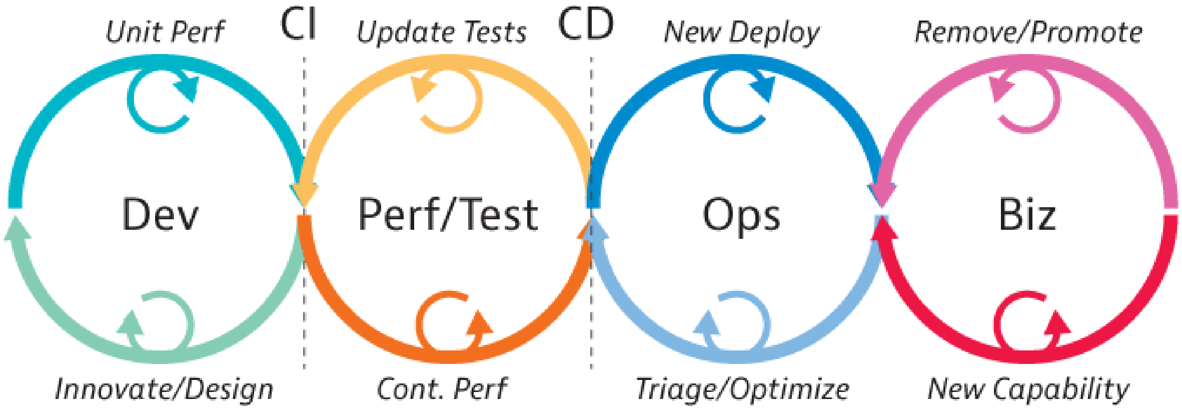
The business benefits include:

- Faster delivery of features
- More stable operating environments
- Improved communication and collaboration
- More time to innovate (and not fix / keep up)

Features of DevOps

- **Source control:** Software developers need to safely store their code and keep track of source-code history and versions. For this reason alone, source control is of critical importance.
- **Issue tracking system:** An issue tracking system allows everyone involved to track current issues, estimates, and deadlines.
- **Build system:** The build system supports continuous integration by building the software, running unit and integration tests, deploying to the integration environment, and performing any other automated checks defined for new versions of the software.
- **Monitoring system:** Monitoring systems continuously track all autonomous systems within the DevOps environment, notifying necessary maintenance staff if a system failure occurs.
- **Communications system:** The constant exchange of information is important so email, wiki, and a real-time chat system being enabled for effective communication and collaboration among all members of the project team.

- **Integration environment: The** integration environment hosts all the virtual machines that make up our DevOps environment
- **Code review system:** To make sure software quality, every line of code being reviewed by a experienced developer. The practice of reviewing code also accelerates career growth and learning.
- **Documentation system:** Regrettably, documentation often remains an afterthought in production software projects. To ensure that documentation being written throughout the project, an automated system being developed to allow developers to write documentation easily, along with source code.



DevOps Goals

- Improved deployment frequency
- To make faster time to market
- Less failure rate to new releases
- Short lead time between fixes
- Improve mean time to recovery

Is DevOps a good career?

DevOps practitioners are among the highest paid IT professionals today, and the market demand for them is growing rapidly because organizations using DevOps practices are overwhelmingly high-functioning to deliver IT services that offer value to the business. According to a study on the application economy and the role of DevOps, 88% of enterprise IT organizations and **LOB (line of business)** executives already have planned to adopt DevOps sometime within the next five years to accelerate delivery of apps and offer customers with higher-quality software. In the last two years, listings for DevOps jobs at **Indeed.com** increased 75 percent. On **LinkedIn.com**, mentions of DevOps as a skill increased 50 percent. In a recent survey by **Puppetlabs**, half of their 4,000-plus respondents (in more than 90 countries) said their companies consider DevOps skills when hiring.

Basic MongoDB Commands:

`db.help()`	get a list of commands
`show dbs`	print a list of all databases on the server
`use myTestDB`	create new database "**myTestDB**"
`db`	know your current selected database
`db.dropDatabase()`	drop the current selected database
`db.createCollection("Employee")`	create new collection "**Employee**"
`show collections`	print a list of all collections created
`db.Employee.drop()`	drop the collection "**Employee**"
`db.Employee.insert({name: 'Raj', address: 'Bangalore'})`	insert document in collection "**Employee**"
`db.Employee.find()`	list the documents in collection "**Employee**"
`{ "_id" : ObjectId("60658a0dbe02cfa1d386ab52"), "name" : "Raj", "address" : "Bangalore" }`	
`db.Employee.update({'name' : 'Raj'}, {$set: {'name' : 'Albert'}})`	update the document in collection "**Employee**"
`db.Employee.find()`	list the documents in collection "**Employee**"

{ "_id" : ObjectId("60658a0dbe02cfa1d386ab52"), "name" : "Albert", "address" : "Bangalore" }	
db.Employee.save({"_id": new ObjectId("60658a0dbe02cfa1d386ab53"), name: "Newton", address: "Delhi"});	save document in collection "**Employee**"
db.Employee.find()	list the documents in collection "**Employee**"

{ "_id" : ObjectId("60658a0dbe02cfa1d386ab52"), "name" : "Albert", "address" : "Bangalore" }
{ "_id" : ObjectId("60658a0dbe02cfa1d386ab53"), "name" : "Newton", "address" : "Delhi" }	

db.Employee.remove({'name': 'Albert'})	delete document in collection "**Employee**"
db.Employee.find()	list the documents in collection "**Employee**"
{ "_id" : ObjectId("60658a0dbe02cfa1d386ab53"), "name" : "Newton", "address" : "Delhi" }	
db.getUsers();	list down all the users of current database
show roles	list down all the roles
db.Employee.dataSize()	get the size of the collection "**Employee**"
db.Employee.storageSize()	get the total size of document stored in the collection "**Employee**"
db.Employee.totalSize()	get the total size in bytes for both collection data and indexes
db.Employee.totalIndexSize()	get the total size of all indexes in the collection "**Employee**"

Docker Commands:

docker --version	get the installed docker version
docker pull hello-world	download the image "**hello-world**" from the docker repository (**hub.docker.com**)
docker images	list all the images that are locally stored with the docker engine
docker run hello-world	create a container from the image "**hello-world**"
docker container ls -a	list all containers
docker container ls -a -s	list the size for all containers
docker rmi 515d5e66f68a	remove the docker image "**hello-seattle**" with image id "**515d5e66f68a**"
docker rm d9bf06498bb2	remove the docker container with container id "**d9bf06498bb2**"
docker history hello-world	display the history of the image "**hello-world**"
docker info	get detailed information about docker installed on the system including the kernel version, number of containers and images, etc.
docker volume create	create a volume which docker container will use to store data
docker volume ls	list all the volumes known to Docker
docker logs c70201336fd8	display the logs of the docker container with contained id "**c70201336fd8**"
docker search hadoop	search for docker image "**hadoop**" on dockerhub
docker network ls	list all docker networks
docker login	login into docker repository (**hub.docker.com**)
docker logout	logout from docker repository (**hub.docker.com**)
docker start c70201336fd8	start the docker container with container id "**c70201336fd8**"
docker stop c70201336fd8	stop the docker container with container id "**c70201336fd8**"

`docker restart c70201336fd8`	restart the docker container with container id "**c70201336fd8**"
`docker inspect c70201336fd8`	get detailed information about the docker container with container id "**c70201336fd8**"
`docker stats c70201336fd8`	get the statistics of the docker container with container id "**c70201336fd8**"
`docker image ls`	List all images that are locally stored with the docker engine.

"While Docker automatically captures logs for you, it does not also rotate them. In fact, currently none of the provided packages set up any log rotation. You'll need to do that yourself in most cases. Rather frustratingly, Docker also does not respond to a signal to tell it to reopen logs. If you send it the standard HUP signal, it will instead restart all the containers, which is not what you want. The current best practice for rotation of Docker logs is to have logrotate use the copytruncate method to copy the logfile and then truncate it in place. There are open bugs against docker asking for a better solution."

— Karl Matthias

`docker system prune`	delete all unused containers, unused networks, and dangling images
`systemctl status docker`	check the Docker service
`systemctl start docker`	start the Docker service
`docker image prune`	remove unused images
`docker save hello-world > hello-world.tar`	save the image "**hello-world**" to a tar archive
`docker load < hello-world.tar`	load the image "**hello-world**" from the saved tar file
`docker export a27999b71e62 > hello-world.tar`	export the docker container with container id " **a27999b71e62**" as a tar archive
`docker import hello-world.tar`	import the contents from hello-world.tar

Linux Commands

The **command-line interface** is one of the nearly all well built trademarks of **Linux**. There exists an ocean of **Linux commands**, permitting you to do nearly everything you can be under the impression of doing on your Linux operating system. Although, this to the end of time creates a problem: by all of so copious commands accessible to manage, you don't comprehend where and at which point to fly learning them, especially when you are learner. If you are facing this problem, and are peering for a painless method to begin your command line journey in Linux, you've come to the right place, as in this, we will launch you to a hold of well liked and **helpful Linux commands**.

Description:

Display system date and time.

Command:

```
date
```

Description:

Display calendar.

Command:

```
cal
```

Description:

Display date, time and calendar.

Command:

```
date & cal
```

Description:

Display August month 2016 year calendar.

Command:

```
cal 8 2016
```

Description:

Used to clear the terminal window.

Command:

```
clear
```

Description:

Exit from the terminal window.

Command:

```
exit
```

Description:

Display free and used system memory.

Command:

```
free
```

Description:

Display free and used system memory in bytes.

Command:

```
free  -b
```

Description:

Display free and used system memory in kilobytes.

Command:

```
free  -k
```

Description:

Display free and used system memory in megabytes.

Command:

```
free  -m
```

Description:

Change user password.

Command:

```
passwd
```

Description:

Power-off the machine.

Command:

```
shutdown
```

Description:

Power-off the machine immediately.

Command:

```
shutdown  -h now
```

Description:

Power-off the machine after 10 minutes.

Command:

```
shutdown  -h +10
```

Description:

Print current working directory.

Command:

```
echo $PWD
```

Description:

Print previous working directory.

Command:

```
echo $OLDPWD
```

Description:

Executes the 11th command in command history.

Command:

```
!11
```

Description:

Reveals your command history.

Command:

```
history
```

Description:

Power off or reboot the Operating system.

Command:

```
sudo reboot
```

Description:

Display the IP address of the host.

Command:

```
ip address
```

Description:

List the size of files and directories.

Command:

```
ls  -s
```

Description:

View mounted file systems.

Command:

```
mount
```

Description:

Display the information of disk usage of files and directories.

Command:

```
du
```

Description:

Tells you how long the system has been running.

Command:

```
uptime
```

Description:

Set current date as 02 Nov 1988.

Command:

```
date -- set 1998-11-02
```

Description:

Set current time as 12:11:02 IST.

Command:

```
date -- set 12:11:02
```

Description:

View and change the configuration of the network interfaces on the system.

Command:

```
ifconfig
```

Description:

Lists all files and directories in the present working directory.

Command:

```
ls
```

Description:

Report the process information.

Command:

```
ps
```

Description:

Display disk usage.

Command:

```
df
```

Description:

Display disk usage in gigabytes, megabytes, or kilobytes.

Command:

```
df  -H
```

Description:

Delete every file and every directory.

Command:

```
rm  -r *
```

Description:

Provides a quick overview of the currently running processes.

Command:

```
top
```

Description:

The system performs an immediate reboot.

Command:

```
reboot
```

Description:

Terminate processes without having to log out or reboot.

Command:

```
kill
```

Description:

Change the current working directory.

Command:

```
cd
```

Description:

Create a new session on the system.

Command:

```
login
```

Description:

List open files.

Command:

```
lsof
```

Description:

List USB devices.

Command:

```
lsusb
```

Description:

Check the status of the network services.

Command:

```
service network status
```

Description:

Start the network service.

Command:

```
service network start
```

Description:

Stop the network service.

Command:

```
service network stop
```

Description:

Restart the network service.

Command:

```
service network restart
```

Description:

Report information about the users currently on the machine and their processes.

Command:

```
w
```

Description:

Display the current directory.

Command:

```
pwd
```

Description:

Displays CPU architecture information (such as number of CPUs, threads, cores, sockets, and more).

Command:

```
lscpu
```

Description:

Displays the number of processing units available to the current process.

Command:

```
nproc
```

Description:

The system performs an immediate reboot.

Command:

```
init 6
```

Description:

Power-off the machine.

Command:

```
init 0
```

Description:

List files by date.

Command:

```
ls -lrt
```

Description:

Report information about storage devices such as hard disks, flash drives etc.

Command:

```
lsblk
```

Description:

Show exit status of previous command.

Command:

```
echo $?
```

Description:

Lists a few useful info commands.

Command:

```
info
```

Description:

Prints current year's calendar.

Command:

```
cal -y
```

Description:

Check the status of all the services.

Command:

```
service --status-all
```

Description:

Display time in hh:mm:ss.

Command:

```
date +%T
```

Description:

Tells when the user last logged on and off and from where.

Command:

```
last  -1 username
```

Description:

Sort files and directories by extension name.

Command:

```
ls  -X
```

Description:

Display the manual for the pwd command.

Command:

```
man pwd
```

Description:

Displays information about running processes in the form of a tree.

Command:

```
pstree
```

Description:

Resets your terminal.

Command:

```
reset
```

Description:

Displays What date is it this Friday.

Command:

```
date -d fri
```

Description:

Displays the size of each individual file.

Command:

```
du  -a
```

Description:

Display information about the Advanced configuration and power Interface.

Command:

```
acpi
```

Description:

Takes you two folders back.

Command:

```
cd ../..
```

Description:

Takes you to the previous directory.

Command:

```
cd -
```

Description:

Displays a list of shell built-in commands.

Command:

```
help
```

Description:

Lists your last logins.

Command:

```
last yourusername
```

Description:

Create a new directory called myfiles.

Command:

```
mkdir myfiles
```

Description:

Remove the directory myfiles.

Command:

```
rmdir myfiles
```

Description:

Disable password for a specific user "root1".

Command:

```
passwd -d root1
```

Description:

Switch to user "root1".

Command:

```
sudo su root1
```

Description:

Exit from the terminal window.

Command:

```
logout
```

Description:

Creates a user "root1".

Command:

```
useradd "root1"
```

Description:

Assign password to user "root1".

Command:

```
passwd "root1"
```

Description:

Repeats the last command.

Command:

```
!!
```

Description:

Display Who you are logged in as.

Command:

```
whoami
```

Description:

Display the login name of the current user.

Command:

```
logname
```

Description:

Report the name of the kernel.

Command:

```
uname
```

Description:

Print the kernel version.

Command:

```
uname  -v
```

Description:

Print the operating system.

Command:

```
uname  -o
```

Description:

Report the machine hardware name.

Command:

```
uname  -m
```

Description:

Print version information and exit.

Command:

```
uname  --version
```

Description:

Print the kernel release.

Command:

```
uname  -r
```

Description:

Report the network node hostname.

Command:

```
uname  -n
```

Description:

Display all port connections (both TCP and UDP).

Command:

```
netstat  -a
```

Description:

Display only TCP (Transmission Control Protocol) port connections.

Command:

```
netstat  -at
```

Description:

Display only UDP (User Datagram Protocol) port connections.

Command:

```
netstat  -au
```

Description:

Display all active listening ports.

Command:

```
netstat  -I
```

Description:

Display all active listening TCP ports.

Command:

```
netstat  -It
```

Description:

Display all active listening UDP ports.

Command:

```
netstat  -lu
```

Description:

Reveal all the information about the current user (user id, username, group id, group name etc.).

Command:

```
id
```

Description:

Reveal all the information about the user "root1" (user id, username, group id, group name etc.).

Command:

```
id  root1
```

Description:

Print the machine's architecture.

Command:

```
arch
```

Description:

Display the list of available fonts.

Command:

```
fc-list
```

Description:

Create two directories (myfiles, files).

Command:

```
mkdir myfiles files
```

Description:

install apache (CentOS).

Command:

```
yum install httpd
```

Description:

install apache (Ubuntu).

Command:

```
apt install httpd
```

Description:

upgrade apache (CentOS).

Command:

```
yum update httpd
```

Description:

upgrade apache (Ubuntu).

Command:

```
apt update httpd
```

Description:

uninstall apache (CentOS).

Command:

```
yum remove httpd
```

Description:

uninstall apache (Ubuntu).

Command:

```
apt remove httpd
```

Description:

Display usage summary for the command (date).

Command:

```
date --help
```

Description:

List active connections to/from system.

Command:

```
ss -tup
```

Description:

List internet services on a system.

Command:

```
ss -tupl
```

Description:

Display all active UNIX listening ports.

Command:

```
netstat  -lx
```

Description:

Display all the active interfaces details.

Command:

```
ifconfig
```

Description:

Display information of all network interfaces.

Command:

```
ifconfig  -a
```

Description:

Compare the contents of two files (1.txt, 2.txt).

Command:

```
diff 1.txt 2.txt
```

Description:

Tells you how many lines, words, and characters there are in a file (1.txt).

Command:

```
wc 1.txt
```

Description:

Compresses file (1.txt), so that it take up much less space.

Command:

```
gzip 1.txt
```

Description:

Uncompresses file (1.txt) compressed by gzip.

Command:

```
gunzip 1.txt
```

Description:

Examine the contents of the file (1.txt).

Command:

```
cat 1.txt
```

Description:

Display calendar.

Command:

```
ncal
```

Description:

Removes the file (1.txt).

Command:

```
rm 1.txt
```

Description:

Rename a file named 1.txt to 0.txt.

Command:

```
mv 1.txt 0.txt
```

Description:

Replace the contents of 0.txt with that of 1.txt.

Command:

```
cp 1.txt 0.txt
```

Description:

Create a empty file (test.txt).

Command:

```
touch test.txt
```

Description:

Print the last 10 lines of a file (1.txt).

Command:

```
tail 1.txt
```

Description:

Print N number of lines from the file (1.txt).

Command:

```
tail  -n N 1.txt
```

Description:

Prints the number of words in a file (1.txt).

Command:

```
wc -w 1.txt
```

Description:

Prints the number of characters from a file (1.txt).

Command:

```
wc -m 1.txt
```

Description:

Prints the length of the longest line in a file (1.txt).

Command:

```
wc -L 1.txt
```

Description:

Print information about usb ports, graphics cards, network adapters etc.

Command:

```
lspci
```

Description:

View contents of a file (1.txt).

Command:

```
less 1.txt
```

Description:

Display calendar (last month, current month, and next month).

Command:

```
cal -3
```

Description:

Compare the contents of three files (1.txt, 2.txt, 3.txt) line by line.

Command:

```
diff3 1.txt 2.txt 3.txt
```

Description:

Compare two files (1.txt, 2.txt) line-by-line.

Command:

```
comm 1.txt 2.txt
```

Description:

Perform byte-by-byte comparison of two files (1.txt, 2.txt).

Command:

```
cmp 1.txt 2.txt
```

Description:

Prints the CRC checksum and byte count for the file "myfiles.txt".

Command:

```
cksum myfiles.txt
```

Description:

Append contents of files (1.txt, 2.txt) into one file (0.txt).

Command:

```
cat 1.txt 2.txt > 0.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt).

Command:

```
sed  r 1.txt 2.txt 3.txt > 0.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt).

Command:

```
sed  h 1.txt 2.txt 3.txt > 0.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt).

Command:

```
sed  -n  p 1.txt 2.txt 3.txt > 0.txt
```

Shortcuts:

```
| ctrl+c          | Halts the current command          |
| ctrl+z          | Stops the current command          |
|                 |                                    |
| ctrl+d          | Logout the current session         |
| ctrl+w          | Erases one word in the current line |
|                 |                                    |
| ctrl+u          | Erases the whole line              |
| ctrl+r          | Type to bring up a recent command   |
```

Description:

Writes contents of a file (0.txt) to output, and prepends each line with line number.

Command:

```
nl 0.txt
```

Description:

Create a empty file (test1.txt) inside a directory (test).

Command:

```
mkdir test
cd test
pwd
touch test1.txt
```

Description:

Gather information about hardware components such as CPU, disks, memory, USB controllers etc.

Command:

```
sudo lshw
```

Description:

Gather information about file system partitions.

Command:

```
sudo fdisk -l
```

Description:

Displays the line (good morning) in which the string (good) is found in the file (1.txt).

Command:

```
grep good 1.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt) using for loop.

Command:

```
for i in {1..3}; do cat "$i.txt" >> 0.txt; done
```

Description:

Search for files (test.txt, test1.txt, test2.txt, test.php, test.html) in a directory as well as its sub-directories.

Command:

```
find test*
```

Description:

Displays status related to a file (1.txt).

Command:

```
stat 1.txt
```

```
###
|     Command      |  Description       |
|:-----------------|-------------:      |
| vi               | Open vi editor     |
| i                | Go to Insert mode  |
|                  |                    |
| a =20; b =64;    |                    |
| print (a + b);   |                    |
| Hit Escape to return to Normal mode.  |
| :w hello.py      | Save text          |
| :q               | Quit               |
| python hello.py  |Print the output:84 |
```

Description:

Download the file (file.txt) from url "http: //website.com/files/file.txt".

Command:

```
wget http://website.com/files/file.txt
```

Description:

Display host's numeric ID in hexadecimal format.

Command:

```
hostid
```

Description:

Display file type of the file (myfiles.txt).

Command:

```
file myfiles.txt
```

Description:

Create a file (myfiles.txt) containing a text (Hello World).

Command:

```
echo 'Hello World' > myfiles.txt
```

Description:

Create a file (myfiles.txt) containing a text (Hello World).

Command:

```
printf 'Hello World' > myfiles.txt
```

Description:

Display IP address of the hostname.

Command:

```
hostname  -i
```

Description:

Add a new line of text to an existing file (1.txt).

Command:

```
echo "Hello world!" >> 1.txt
echo "this is 2nd line text" >> 1.txt
echo "last line!" >> 1.txt
```

Description:

Displays a single line description about a command (cal).

Command:

```
whatis cal
```

```
###
|     Command      |  Description       |
|:-----------------|-------------:      |
| vi               | Open vi editor     |
| i                | Go to Insert mode  |
| Type some text.  |                    |
| Hit Escape to return to Normal mode.  |
| :w test.txt      | Save text          |
| :q               | Quit               |
| :q!              |Quit without saving |
```

```
###
|     Command      |  Description          |
|:-----------------|-------------:         |
| vi               | Open vi editor        |
| i                | Go to Insert mode     |
| $name = "Paul";  |                       |
| print "$name";   |                       |
| Hit Escape to return to Normal mode.     |
| :w hello.pl      | Save text             |
| :q               | Quit                  |
| perl hello.pl    |Print the output: Paul |
```

```
###
|     Command                |  Description           |
|:---------------------------|-------------:          |
| vi                         | Open vi editor         |
| i                          | Go to Insert mode      |
| echo "What is your name?"  |                        |
| read PERSON                |                        |
| echo "Hello, $PERSON"      |                        |
| Hit Escape to return to Normal mode.                |
| :w hello.sh                | Save text              |
| :q                         | Quit                   |
| sh hello.sh                | Output:                |
|                            | What is your name?     |
|                            | If you enter: Zara Ali |
|                            | Hello, Zara Ali        |
```

Description:

Check the network connectivity between host (your connection) and server (Google server).

Command:

```
ping google.com
```

Description:

Find the location of source/binary file of a command (cal).

Command:

```
whereis cal
```

Description:

List the files in the bin directory.

Command:

```
ls /bin
```

Description:

List the files in the bin directory and the etc directory.

Command:

```
ls /bin /etc
```

Description:

Moves the file test.txt to the folder newrepo.

Command:

```
mv test.txt ./newrepo
```

Description:

Deletes all the lines in the test.txt containing tue word.

Command:

```
sed -i "/tue/d" test.txt
```

import subprocess
subprocess.call ('**linux command**')
	import os
	os.system('**linux command**')

import os
os.system('ls')
	List all the files and directories in the current directory

import subprocess subprocess.call ('ls')	

Linus Benedict Torvalds is a Finnish-American software engineer who is the creator and, historically, the main developer of the Linux kernel, used by Linux distributions and other operating systems such as Android and Chrome OS.

References:

- **The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations** By Gene Kim
- **DevOps For Dummies** By Emily Freeman
- **The Future of DevOps** By Tom Smith
- **Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale** By Jennifer Davis